

December 5, 1987

ST INDEX 13

HOOK UP ANOTHER DRIVE

From:MIKE MILLER
 Subj:SECOND DRIVE

1.....	Read Data.....	30
2.....	Side 0 Select.....	32
3.....	Logic Ground.....	15
4.....	Index Pulse.....	8
5.....	Drive 0 Select.....	10
6.....	Drive 1 Delect.....	12
7.....	Logic Ground.....	17
8.....	Motor On.....	16
9.....	Direction In.....	18
10.....	Step.....	20
11.....	Write Data.....	22
12.....	Write Gate.....	24
13.....	Track 00.....	26
14.....	Write Protect.....	28

* _ * _ * _ * _ * _ * _ * _ * _ * _ * _ * _ *

I've used both a 5 1/4" 720K Teac disk drive and a 3 1/2" 720K Toshiba disk drive as drive 'B' on my new ST.

Hardware Specification

The base machine outline specification is as follows:

T800-20 Transputer 10MIPS, 1.5 Mflop
Three 20Mhz links, buffered 4Mbyte DRAM
1 Mbyte dual-port video RAM Colour blitter
True DMA SCSI port for 40M (minimum) hard disc
Three internal expansion slots
68000 Mega ST as I/O processor (plug in card connects fourth 20Mhz link)

Screen Resolution and Use

The table below lists the screen resolutions and their probable typical use. All the following are at 60Hz with portrait orientation.

Mode	Resolution	Width	Description
0	1280 x 960	4 bits/pixel	4 bits/colour or monochrome (Desk Top Publishing, engineering drawings)
1	1024 x 768	8 bits/pixel	8 bits/colour (CAD, colour pictures, graphs)
2	640 x 480	8 bits/pixel	8 bits/colour 2 screens (Animation)
3	512 x 480	32 bits/pixel	24 bits colour, 1 overlay bit, 7 tag bits (True colour, smooth shading, 3D modelling)

The Blitter

The Perihelion blitter is based on work done by Dr Phil Willis of the University of Bath. It provides meaningful operations with colour and colour look-up tables (CLUTs) and implements very fast 2-D raster graphics operations,

such as fast font drawing. It also provides a 32-bit wide pipeline (with four tests on each of eight pixels concurrently), and is synchronised with blanking.

Using the blitter, square area fill takes 128 megapixels per second, arbitrary two colour character drawing takes up to 64 megapixels per second, and full 2-D

block copy takes 16 megapixels per second.

Expansion Capability

The Perihelion design provides for three expansion cards within the box. These can be memory cards, providing a maximum of 64Mbytes using 4M parts, or various

versions of alternative graphics cards. The full transputer bus is brought out so any type of peripheral may be connected.

The expansion sockets also bring out the transputer links and control signals. This means that cards containing extra transputers can be added, and the size of the cards allows for four transputers with up to 1Mbyte of RAM each on a single card. One workstation can therefore contain 13 processors. Other link connections can be made outside the box to parallel processor farms of multiple

processors. The link connections can also be made to fast peripherals such as a laser printer or disc server.

The Transputer

The T414 is a 32-bit processor that consists of a RISC style CPU, 2K of fast on-chip RAM, an external memory interface and four serial links which may run at 5, 10 or 20 Mbits/second. The T800 is similar except that it also contains a floating point processor and 4K of RAM.

The programmer's model consists of a three register evaluation stack, a workspace pointer and an instruction pointer. A small number of instructions exist for loading and storing values on the stack and for altering the flow of control, the remainder operate on operands on the stack.

The processor has microcoded support for processes at two priority levels. High priority processes may preempt low priority processes after any instruction and run until they give up the processor. High priority processes are essentially equivalent to interrupt routines on conventional processors. Low priority processes are round-robin scheduled on a timesliced basis. Timeslicing only occurs on particular instructions which are defined so that the minimum of state need be saved; process switching is therefore very fast.

The transputer achieves inter-process communication through channels, which are single words of memory. Two processes that wish to communicate rendezvous at a channel and exchange data by copying from one buffer to another. As this is implemented by the microcode, the cost of copying lies only in the memory accesses for the data and not in instruction fetches. Communication is strictly one-to-one and channels may not be shared by more than one sender or receiver. The inter-processor links are designed to behave exactly like channels, and are used with the same instructions.

Parallel Programming

The unique aspect of the Atari/Perihelion design is that it provides multiple processors within a single workstation. The use of multiple processors means that it is possible to write application programs which make use of the possible parallelism inherent in such systems.

Application programs can run under Helios using three programming philosophies.

The first of these is the traditional programming model. A program can be taken from another environment, such as Unix or a PC, and with little or no change converted to run under Helios. C and the Unix C library is provided, and such programs will run as a single process in the machine.

Other programs, again probably from Unix, will run in several sections all of which may be run in different processes and connected by pipes. Helios encourages the use of many small programs which work together to create a final

product. A common example is a pre-processor, a compiler front end, a compiler back end, an assembler and a linker. These can all be run together with intermediate connections made by pipes. Under other operating systems the different processes are timesliced on the one single processor. Under Helios these different processes can be allocated to different processors, so that the individual parts actually run at the same time.

This type of "per-process" parallelism is easily understood, and many applications are already in this form. Examples include a word processor with background spooling and spelling checking or background jobs such as message systems or archiving. If an application is being altered then the use of extra processes should be kept in mind.

The final way in which parallelism may be exploited is by the use of parallel algorithms. These tend to be hard to find for programmers used to the sequential nature of normal computers, but a look at the real world shows, of course, everything running in parallel. Applications using parallel algorithms will normally be written from scratch with such ideas in mind. The benefit is that such programs will run much faster when the user provides more power in the form of more processors. Many examples of parallel algorithms exist, such as ray tracing, spreadsheet calculations, even compImplementation

Helios presents a low level interface that should be familiar to programmers who have worked on Unix. Each user runs a number of tasks which can communicate

between themselves using a simple message passing protocol. A message may be transferred between two tasks in the same machine or between tasks in different

processors; in each case the call is identical and the message is copied rather than passed by reference.

Each task is constructed from a number of interconnected transputer processes which can communicate either by message passing or by sharing data. Different tasks may be written in different languages as all communication at this level is by message passing.

The design of Helios is based on the client-server model, where application tasks request services from system provided server tasks. These server tasks may be present in any or all of the processors available, although each processor must run the bare minimum of the name server which identifies the location of other services. Other servers include file handlers, window managers, date servers, spoolers and so on. All servers respond to a general server protocol which is designed so that servers may be stateless and hence unaffected by crashes and communication losses. This mechanism allows a wide choice in the way in which servers are implemented; for example, floppy discs will be written using the MS-DOS format while winchester discs will use a format similar to Unix.

As noted above, the transputer does not contain any memory management unit and none may be added externally. Helios assumes that each processor is allocated to a single user, and protects processors by a capability mechanism. An access matrix is used to implement a filing system control scheme.

System Tools

Helios is written in a mixture of transputer assembler and C. System tools include a macro preprocessor, a C compiler, an assembler and a linker. In fact these last two items are the same program as the transputer has a variable length instruction set that requires a 'code growing' algorithm to ensure optimal code lengths sequences. This must be performed in the linker once all external references have been satisfied.

A debugger that allows one transputer to investigate another will also be included in the package (although not in Version 1.0). Third party software developers are providing Pascal, Fortran, Lisp and BCPL as well as the

traditional transputer language occam.

User Interface

The user interface consists of two parts: a command line interface similar to the Unix C-shell which provides the usual commands such as grep, ls, more, and so on; this will be coupled to an implementation of Xwindows V11 to provide the now familiar windowing mechanism.

The system will appear similar to a more conventional machine, except that when the commands are piped together the operating system may make use of more than one processor to run the commands concurrently rather than timesliced as in a single processor environment.

The programmer's interface will be complemented by a 'point and push' graphical interface for the less experienced user. This will use a mouse and pull-down menus and will be implemented on top of Xwindows.

SPECTRUM 512 Picture File Format

There're two kinds of Spectrum files: uncompressed (.SPU) and compressed (.SPC).

The .SPU file is always 51104 bytes long and consists of two parts: first the bit map (32000 bytes) and then the color map (19104 bytes). Because SPECTRUM 512 cannot display the topmost raster line, you have 199 lines instead of 200. For that reason, the top line in the bit map (the first 160 bytes of the file) contains no useful information and you should simply ignore it (SPECTRUM 512 fills it with zeros). Otherwise the bit map has the usual meaning -- low res, 4

planes, 160 bytes per raster line, giving each pixel a certain color number from 0 to 15. When you want to determine the color of any particular pixel in the .SPU file, you should first find it's color number (using Get Pixel \$A002 rom A-line, for example; or your own routine).

To find the actual color of the pixel, look at the color map portion of the file. It consists of 199 blocks, one 96-byte block per raster line (no dummy top line here!). Each block contains 48 word color values in the usual ST format (00000rrrr0ggg0bbb, r=red, g=green, b=blue). You will find the block you

need for your particular pixel at the address $(Y-1)*96$ from the start of the color map, where Y is the Y coordinate of your pixel, $1 \leq Y \leq 199$.

Finding which of the 48 colors in the block is the color of your pixel is a little trickier. To do that you'll need the color number that you found from the bit map and the X coordinate of your pixel. First, multiply the color number by 10. If the color number is even, add 1 to the result; if it's odd, subtract 5. Let's call the resulting number X1. Now, if X is less than X1, leave the color number as it is; if X is more or equal X1 but less than X1+160,

add 16 to the color number; and if X is more or equal X1+160, then add 32 to the color number. This adjusted color number (it could be anything from 0 to 47) is the solution to our problem! It shows which of the 48 color values in the block corresponds to your particular pixel.

An example:

X=139 and the color number is 3.
 Multiply by 10: $3 \times 10 = 30$.
 Subtract 5 (because 3 is odd): $X1 = 30 - 5 = 25$.

Okay, 139 is more than 25, but less than 185 ($25 + 160$). So, we add 16 to the color number and the result is 19. That pixel's color will be found in the word number 19 of the block (word count starts at 0).

.SPC files consist of 3 parts: 12-byte header, compressed bit map and compressed color map. Here's the header layout:

word	\$5350	("SP")
word	0	for future enhancements <>0 means there're additional records in the file, following the compressed color map
long	length of the bit map	
long	length of the color map	

Bit map compression is a slightly modified RLE. Each record consists of a header byte followed by one or more data bytes. A positive header ($0 \leq n \leq 127$)

means copy the next $n+1$ bytes literally, a negative ($-128 \leq n \leq -1$) - copy the next byte $-n+2$ times (minimum 3; the encoder does not compress if there're just

2 equal bytes in a row). First goes bit plane #0, all scan lines from 1 (not 0!) to 199, without breaks at the ends of scan lines (only at the end of the last line), then bit planes #1,2 and 3. Terminate decompression when the number

of bytes extracted reaches 31840 (4 bit planes x 199 lines x 40 bytes per line). The length of the compressed bit map is always even, so there might be one filler byte at the end of it. If you convertan .SPC file to .SPU to display

it on the 512-color screen, fill scan line #0 with zeros (first 160 bytes of the .SPU file) to give it the same color as the top screen margin.

In the color map each short 16-word block is compressed separately (there're 3 of them in each scan line, 597 altogether). The compressed record for each short block starts with the bit vector (1 word) indicating which of the 16 color slots are used, followed by the color values themselves (each is a word).

For instance, \$000A,\$0707,\$0777 should be expanded into 0,\$0707,0,\$0777, twelve zeros (\$000A = %0000000000001010, meaning only colors #1 and 3 are used). Unused

color slots should always be filled with zeros. Color #15 is never used in a finished Spectrum picture, so bit #15 of the bit vector must always be 0. Color

#0 is always used (as a background) and it's always black. Since it makes no sense to have this extra 0 in every compressed record, bit #0 of the bit vector

is always cleared, indicating to the decoder to fill slot #0 with 0. The number

of color values following the bit vector in the compressed record is equal to the number of 1's in the bit vector.

If the above compression rules are observed the length of the compressed bit map will never be greater than 32092 bytes, and the length of the compressed color map will never be greater than 17910 bytes. Spectrum rejects SPC files that exceed these size limits.

TECHNICAL HELP

And now a better way!!!

by Mr. Goodprobe

It appears that for every good and needed idea mankind has come up with, someone has added some extra features and labor saving additions that make that

device the "thing you can't live without!" Such an item is this little null modem that will allow you to transfer files from your St to your 8 bit Atari, and to other computers as well. Now I fully well realize that Antic published and excellent article on this very topic a while ago, but their procedure forced the user to have in his possession an interface of some sort for his 8 bit Atari, whether it be a PR Connection, 850 or the like, nonetheless, you had

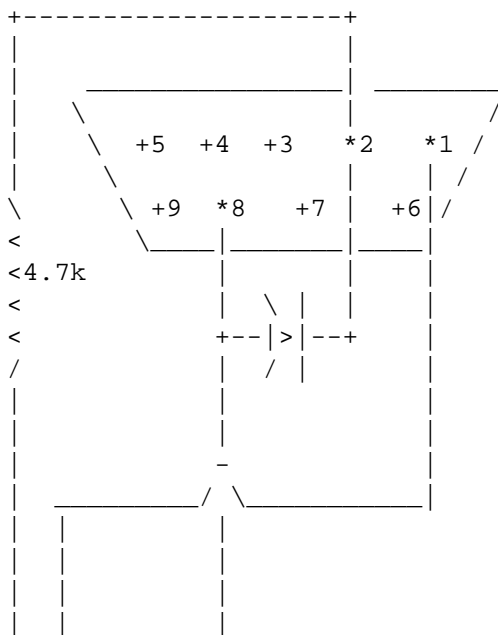
to have this item in your possession or the transfer of data between both computers was impossible. But now I will in a few short paragraphs show you how

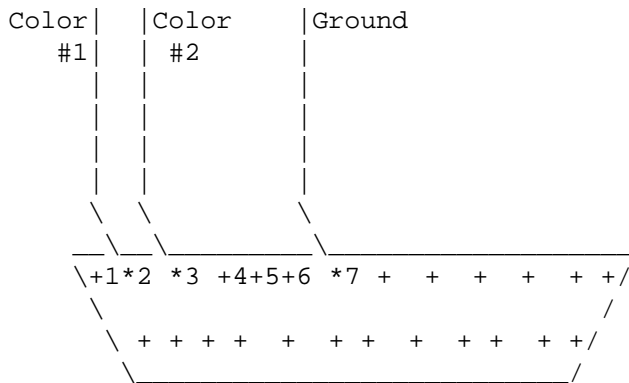
to accomplish the very same feat without an interface.

What we will be assembling is a skimpy at best data transfer setup that is easy

on the olewallet. First you will need:

- 1.) A joystick cable such as used on your favorite alien blaster! This must be complete with the plug on one end.
- 2.) A 25 pin connector to plug in the modem port in the rear of your ST.
- 3.) One 3.9k to 4.7k 1/2 watt resistor 10% tolerance
- 4.) A short run of 3 conductor wire
- 5.) A diode (such as ECG 109 or any small switching signal diode).





As you can see, we are merely connecting the receive data line to the transmit data line on the ST, while at the very same time connecting the transmit data line of the 8 bit Atari to the receive data line of the ST. You will then need to use a term program on the 8 bit that addresses the joystick port, such critters are MPP term, and also the term which came with the Supra series of 1200 baud modems. You can go as fast as the 8 bit program allows, with a ceiling of 9600 baud if someone were to implement faster transfer routines in the aforementioned programs. I have been using this type of setup for several months now...and its a dream!!!

This same basic setup can be used to connect the 8 bit Atari to most any computer out there on the market today, with no difficulties whatsoever.

By the way, you might find it amazing that this entire article this week has been typed, edited, spell-checked and saved to disk..while downloading a 500+ K

disk of Music Studio files!! "Sounds like multi-tasking to me" you might comment, and guess what? You are right! I am using a new file protocol .TXF for

Interlink called MultiXY, it allows you to download/upload in the background while running your favorite program and doing other more useful things! I started the transfer, and then went to my file menu on Interlink, pulled it down, and clicked on "Execute". I then chose my Word Writer program and loaded

it from there, and thats were we are at this very moment! I am able to check the progress of my xmodem or ymodem file transfer by going up to the accessory area of my screen, and viewing the small display located there. It tells me how

many blocks have been transfered, and with how many errors. If you show a file,

and save to the disk, you get an error, but of course your BBS system will resend that block...Amazin what they are doing now eh? This accessory only works with Interlink 1.71 and up, so unfortunately owners of other programs cannot enjoy this great time saving feature...

I will be doing a much more thorough review of this handy little file transfer protocol in the next issue or two of Zmag.

Keep those Atari's hummin!
Mr.Goodprobe
(on lend from) Midtown TV
Atari 8/16 Sales/Repair

Please keep those project ideas coming, many interesting items shall appear in futre issues due to the gracious response of our readers! Good job!

GARBAGE ON THE LINE

by Calamity Jane --CJ--

...Life On The Frontier...

or

--Roughing-It-Easy--

I am Calamity Jane, OpSys of The Prairie Chip BBS in Wyoming. All of that is just a coincidence... really!! Do you think I planned that? That my life is THAT organized... hardly. Ever heard of Wyoming (Wi-O-ming)?? Where the sidewalks end and the West begins and the trail cuts across the lonely prairie.

The fierce hositlities of the Araphoe, Sioux, Shoshone, Cheyenne, and some Ute, have kept our population down, but yes, real people do live here in our many thousands of acres of rolling plains. And we -DO- own computers...

This article has been inspired by my BBS friends in New York City. <hi! guys> I had a message very typical... "So YOU are the one who lives in Wyoming!" very

funny... I seem to get a 'hard-time' for being from Wyoming where ever I go... and I have come up with several defense mechanisms... I proceeded to hint on how I run my BBS in such a remote frontier without the usual luxuries of electricity. "How do you read? or watch TV?" I was asked. I mentioned I don't watch much TV, and read by the glow of the monitor. They were intrigued... Thus the reason for this article.

There is such a combination of the old ways and the new in my life, and they are combined in such a way that makes my life quite tolerable, _pure and I don't have millions of people around me. Oh, give me a home where the buffalo roam... Yes, the millions of buffalo are gone, but there are a few, and I can count on a small <but rather exciting> STampedede on the average of every couple of weeks. They do, however, continually knock the pole over and pound the open

wire <that is the telephone line> into the ground in a cloud of duST. No optic fiber here.

I live in a log cabin. The dictionary calls it a 'small house... rudely constructed. Nothin' rude here, we're friendly folk. It has all the comforts of home. I purify my own water, I use an outhouse, I cook on a woodstove. It supplies warmth and gives me something to constantly be doing. <NO, not the outhouse, the stove !!> Cutting the trees, chopping the wood, hauling the wood,

loading the stove, emptying the stove... you get my idea. I burned off my eyebrows and eyelashes once when I poured a dab of kerosene on what I THOUGHT was a dead fire. However, they grew back. I have a beautiful brand-new washing machine that never needs repair. It's a stick. The motion of the water in combination with soap carries the dirt away. I don't have to listen to the hounding of my clothes either. I have a rare refrigerator that runs on kerosene, few exist though. I have learned to live without the need for everything electric. "Less Power to You !" I not only know how many Desk Accessories you can have per disk, or to forget extended format, but that a kerosene lantern with a 1" wick will burn apx. 45 hours on 1 quart of kerosene at the rate of 5 hours each day <12 gal. a year>. I use tallow rather than wax

candles as they burn longer, are brighter, and fairly smoke free. I get 48 hours out of a 1" by 9". They are also free, if you make them. I do not use oleman lanterns as they hiss, clank & blind me, just like civilization. As you can see, electricity is the least of my concerns. Till I bought the

computers. Then it became a major concern. But the power lines just ain't reached way-out-here yet !! The wind does blow, mighty hard too; thus the source of power that keeps me in CHAT. I am ala natural, on the great treeless stretches, which roll away as far as the eye can see.

The wind-generator, heck I still call it the windmill, is a noisy, clanky, cus and if the wind doesn't blow (hah!) the batteries won't charge. My first concern is the wind will blow too hard and blow it up. How do I surge protect THAT ?? Still it does keep me up and running and in enough power to keep The Chip waiting for you at 2400 bps. I have considered many ways to gain access to the power to run my computers. I am considering and finding out about high output silicon solar cells and other such solar devices, but right now I'll stick with the good ol wind, and the slow-revving, big fly wheel, last forever,

donkey engine. The storage of this energy is a constant pain. It is known as Wind Generator System Storage Problems. The batteries are still a bit of a black art even in this high tech age. The batteries are the common lead acid type used in cars and will last several years. About the only thing I must do to them is check them daily and feed them rain water if needed.

I got tired of traveling 13.2 miles to the nearest neighbor to 'chat', so I talked long and hard... Smooth taklin' Joe finally ambled into town to see if they had any more of "those damn ST machines". Now we chat in comfort without the worry of seeing the elephant, Indian attacks, fierce storms or snakes. I never know when I will hear the war cry and a cloud of arrows. molossi tellim piduui. It can be on the average of once a week. The phone lines require regular maintainence between the Indians and the buffalo. The Indians love the

colored glass from the insulators, so I build boxes to cover them, & paint owls on them <owls are bad luck to an Indian>. Quite often I must pour a bucket of water on the ground rod, this makes for a better connection...you know, less line noise. It's hell running a 24 hour BBS on an eight member party line. It ain't too easy on them either!! Eight of the beST callers any SysOp could want.

I can still see the ruts on the land from the conestoga wagon's. There is something about the wide open spaces, spacious skies and more than 50 miles to anything!! Try that, in NYC, Detroit, or D.C. Winter is a challenge in itself,

something coming on here quite rapidly to stay. The geese are flying south, the beaver lodges have more logs in them, the squirrels tails are bushier than usual, the bark on the trees is thicker, animals gathering their food supply early, and FoReM SysOps all over are getting their DOORS up and secured. Must gonna bea long hard winter. I can finally learn EMACS. I must get to the wood supply... <I find the one-woman cross cut saw nice.> It's quiet, keeps me realistic about being a wood hog, keeps my canoe arm in shape, but can be someone hard to start in the cold weather...hahaha. Because things tend to cool

off in the cabin at night, I have specially built insulated covers for the equipment and the disks... Got tired of icicles forming on the SC1224. The Inside-Out Room for the BBS was built in a different way as to prevent such a drastic change in temperature. Completely climate-controlled. Solar-heated, as wood ash is hell on computer equipment!! I'll bet The Chip is one of the most comfortable BBSs in the land.

Well... the night is late, the fire is low, time to hit the hay <no really I have a waterbed... I use the heat from the manure pile to heat it>. The cat wants out <he's a 85 pound cougar> and the wolf wants in. Brave the hardships

and dangers of the unknown wilderness. Call The Prairie Chip BBS. "Where Men are Men and Women are....." 3/12/2400 bps, 30-45 mph wind, 307-635-0148. ST FoReM 2.0... Wrap your feet around your power supply, and stay warm beside your modem. You may not come back alive!!

--Happy Trails--

Ifins you want... permission granted to reprint...

--CJ--

Xx IBM EMULATOR UPDATE

FROM ZMAG JUNE 1987

Business Users 16-Jun-87

Sb: Info on PC-Ditto

Fm: David and Sandy Small 76606,666

To: Pc-Ditters

I spent awhile on the phone with Avant-Garde (like three hours) talking about PC-Ditto, so I thought I'd pass some info along, since there's a lot of curiosity about it.

Jerry called me up from Comdex and said it was amazingly good. He'd done all the things like under the table for a PC or inside for an 8088; it's a software

emulator, all right. Then, he got a bunch of his own software to test it on.. and lo and behold, it worked. He ran LapLink and DesqView on it, and while he says it's slow, it does work. Apparently the emulation is at the chip level.

The guy who did it is an ex-IBM'er and sales type -- he was a pro salesman, believe me -- who's working on it with his wife. Mom and pop shop, so to speak.

They've had lots of marketing offers and are very very busy. Don't expect to get through on the phone, it's always busy.

My own experience with emulators is that you lose a 4X in clock speed with necessary overhead. So I'd expect about a 1-2 Mhz IBM out of this. The good thing is that I/O is done at 68000 speed, 8 Mhz, so it will only appear slow when you get computation-bound..like recalcing a spreadsheet. Burt just copying

files, etc, it ought to scream along pretty nicely.

The big thing that they told me was they wanted everyone to know they weren't connected at all with the MS-EM people. Apparently they had received mucho negative feedback on that other product.

They've tested and certified it with lots of different IBM products; they went through a top-40 IBM product list and it ran all that stuff. The guy who's doing it sounds technically competent and is a good salesman.

Caveat: this is all based on a telephone conversation, but he does know what he's doing with emulators, and it's for real. I'd suggest getting them online on CIS a.s.a.p. to answer questions. It sounds like it'll be a good product.

-- Dave

MIDI

....Michigan Midi Music Madness....

The following article was first published in the Michigan Commodore Users Group newsletter called "SPRITE" in the May 1987 issue.

Musings about Michigan Midi Music Madness by Mike Gillie Copyright 1987.

From time to time (or maybe just one time) I will be writing about my discoveries in the world of MIDI (the Music Instrument Digital Interface). A MIDI interface is available for most popular personal computers. This past November (1986) I slowly began the process of configuring a MIDI system.

I will be presenting MIDI from an amateur/hobbyist perspective. An attempt will be made to convey accurate information. For more information, I would refer the reader to the book "MIDI for Musicians" by Craig Anderton (I purchased my copy at MicroWorld). I have been a subscriber to Electronic Musician (formerly Polyphony magazine) for over 10 years, and have found it helpful. Keyboard is another magazine that covers MIDI.

A computer can serve many purposes in a MIDI system because a computer can be programmed. I am using Music Studio from Activision to run the computer.

Music Studio will work with an Amiga or an Atari ST. Music Studio turns the computer into a Sequencer/ Recorder.

A sequencer is a device that provides a sequence of musical notes. It is a sort of electronic player piano roll. A recorder is like a tape recorder. It "records" the notes played on the keyboard.

Music Studio does a good job of being a sequencer, however it does a poor job of being a recorder. Music Studio can "record" the notes although it won't record the duration of the note. There are other software products that will do this.

Music Studio will present you with a musical staff. You can place the notes on the staff using a mouse (pointing device). On an Atari ST you can play 3 voices at once and on the Amiga 4 voices. When Music Studio is used with MIDI it can control 15 channels of information (and many voices).

The Channel Number, Program Number and range of notes can be assigned to each of the 15 "instruments" Music Studio uses. Each of the 15 instruments can be named. An instrument on the MIDI network can be programmed to recognize only a certain channel. You could have electronic drums on Channel 3 and an electronic piano on Channel 4.

Each device would only recognize data on it's channel. An electronic piano may have many different sounds available. This can be changed by switching to a different program. The computer, acting as a sequencer, can automatically change the programs (sounds) during the song.

In order to use Music Studio with MIDI, you need some kind of musical instrument. Musical hardware is more expensive than computer hardware because the market is smaller. At the low end of the market is the Casio CZ-101. Music Studio has some files that are programmed for this instrument.

The Casio CZ-101 has a mini-keyboard and is a multi-timbral instrument.

"Multi-timbral" means that it can play many timbres (sounds/instruments) at the same time. The Casio CZ-101 can play four different instruments at the same time.

The nice thing about MIDI is that you can now devise a musical system using components. An interesting device is the Yamaha FB-01, which is in the same price range as the Casio CZ-101. The Yamaha FB-01 has no keyboard! However it can play eight instruments at the same time. The FB-01 is ideal for people who can't play the keyboard very well or who need a second synthesizer.

If you have a Casio CZ-101 (or any other MIDI device with a keyboard), it can control the FB-01, as well as itself, giving 12 voices of music. The computer along with Music Studio can control both machines.

I'll postpone a review of the FB-01 until a future issue. If you are considering getting into MIDI, take a good look at the Yamaha FB-01, which I consider a better value than the Casio CZ-101.